REMARKS

Claims 2-3 have been cancelled. Claims 1-2, 4, 7-8, 12-27, and 29-32 are pending. Claims 1, 4, 7, 16, 18, 21, and 29-32 have been amended.

In the Office Action mailed July 28, 2005, the Examiner allowed claim 20. The Examiner rejected claims 1-4, 7-8, 12, 19, 21-24, and 29-32 under 35 U.S.C. § 102(e) as anticipated by U.S. Patent No. 6,098,064 to Pirolli et al. (hereinafter "*Pirolli*").

The Examiner rejected claims 13-14, and 25-27 under 35 U.S.C. § 103(a) as obvious in view of Pirolli and U.S. Patent No. 4,490,782 to Dixon et al. (hereinafter "*Dixon*"). The Examiner rejected claims 15-18 under 35 U.S.C. § 103(a) as obvious in view of Pirolli and U.S. Patent No. 6,453,389 to Weinberger et al. (hereinafter "*Weinberger*").

REJECTION OF CLAIMS 1-4, 7-8, 12, 19, 21-24, and 29-32 UNDER 35 U.S.C. § 102(e) IN VIEW OF PIROLLI

The Examiner rejected claims 1-4, 7-8, 12, 19, 21-24, and 29-32 under 35 U.S.C. § 102(e) in view of *Pirolli*. Applicant respectfully traverses this rejection.

Anticipation under 35 U.S.C § 102(e) requires that each and every element of the claimed invention be disclosed in the prior art. *Akzo N.V. v. United States International Trade Commission*, 1 USPQ 2d 1241, 1245 (Fed Cir. 1986). Further, anticipation requires the disclosure in a single prior art reference of each element of the claim under consideration. *W.L. Gore & Associates v. Garlock, Inc.*, 220 USPQ 303, 313 (Fed Cir. 1983). Applicant respectfully asserts that every element of the present invention is not disclosed in a single prior art reference.

Claim 1

As required by amended claim 1, *Pirolli* does not disclose prefetching of data elements into a Least Recently Used (LRU) cache. In addition, *Pirolli* does not disclose intercepting a request for a data element from a stream of I/O data requests sent from a host and addressed to an LRU cache.

Amended claim 1 recites a Least Recently Used (LRU) cache. An LRU is typically designed to store random access data, not sequential data. When a new data element is added to a full LRU cache, the LRU cache replaces the least recently used data element with the new data element. Specification page 3.

The claimed invention enables efficient storage of sequential data in an existing, unmodified LRU cache by modeling the data elements currently stored in the LRU cache and utilizing the model to decide when to prefetch data elements into the LRU cache. The model handles sequential data in addition to random access data. Consequently, one benefit of the present invention is that it uses an existing, unmodified LRU cache to store sequential data by prefetching sequential data into the LRU cache even though the LRU cache is designed to store random access data. Specification page 5.

In contrast, *Pirolli* does not disclose the use of an LRU cache. Instead, *Pirolli* uses a cache manger that stores and retrieves data from cache memory. *Pirolli*, Col. 4, lines 24-25. The cache manager does not manage the cache memory according to a least recently used scheme. Instead, when the cache memory is full, the cache manager compares a need probability for each document in the cache with a threshold to determine which documents can be deleted from the cache memory. Documents with a need probability below the threshold may be deleted to make room in the cache for new documents. *Pirolli*, Col. 4, lines 24-64.

The cache manager and cache memory in *Pirolli* do not behave according to an LRU scheme. An LRU cache, when full, simply deletes the least recently used data element to make room for a new data element. An LRU cache does not utilize probabilities or thresholds. Specification page 3. Consequently, the invention disclosed by *Pirolli* cannot utilize an LRU cache, as required by amended claim 1. In fact, *Pirolli* teaches away from the use of an LRU cache by requiring the use of a needs list, a need probability and a threshold to determine which documents may be deleted from the cache instead of relying on the least recently used approach to deleting data elements from a cache that is necessarily implemented by an LRU cache.

A host, according to amended claim 1 of the present invention, sends I/O requests for data elements to a cache instead of sending I/O requests directly to a memory device. The present invention intercepts these requests. Specification page 11. If the cache does not contain a requested data element, the cache requests the requested data element from a memory device.

In contrast, *Pirolli* teaches a client computer that addresses and sends document requests *directly to a server*. A cache manager may intercept these requests and prevent them from reaching the server. However, the client computer is unaware of the cache manager and the interception that the cache manger performs. Thus, *Pirolli* does not teach intercepting requests sent from a host and addressed to an LRU cache. In fact, *Pirolli* teaches away from requests addressed to an LRU cache. Pirolli teaches requests comprising a protocol, the name or address of the *server* on which the document is located, and the local name of the document. *Pirolli*, Col. 6, lines 25-30.

Anticipation requires the disclosure in a single prior art reference of each element of the claim under consideration. Since *Pirolli* teaches away from the use of an LRU cache, and teaches requests addressed to a server instead of to an LRU cache *Pirolli* does not anticipate amended claim 1.

Claims 4, 7-8, 12, and 19 depend directly or indirectly from amended claim 1. For the reasons explained above, Applicant respectfully asserts that claims 4, 7-8, 12, and 19 are allowable for at least the same reasons as amended claim 1. In addition, each of these claims is allowable for the following reasons.

Claim 4

Claim 4 requires that the LRU cache be a substantially unmodified, native, LRU-only cache. A substantial benefit of present invention is that it may use a pre-existing, native LRU-only cache without modifications. This allows a user to take advantage of the performance improvements offered by the present invention without having to replace an existing LRU cache being used in a network.

As discussed above, *Pirolli* requires the use of a cache manger that utilizes a needs list, a need probability and a threshold to determine which documents may be deleted from a cache. *Pirolli*, Col. 4, lines 24-64. Once the cache manager determines that a document must be deleted, it deletes the document from the cache. *Pirolli*, Fig. 8. These requirements are incompatible with a substantially unmodified, native, LRU-only cache because this type of cache makes delete decisions based on a least recently used criteria, not on a needs list, need probability, or threshold. Furthermore, a substantially unmodified, native, LRU-only cache is not capable of receiving a delete command from an external cache manager. A particular data elements remains in a substantially unmodified, native, LRU-only cache and is not deleted until the cache fills and the particular data element is the least recently used data element in the cache.

Therefore, because *Pirolli* fails to teach or describe a substantially unmodified, native, LRU-only cache as recited in claim 4, Applicant asserts that claim 4 is in condition for allowance over *Pirolli*.

Claim 7

Pirolli does not anticipate claim 7 because *Pirolli* does not disclose modeling an LRU cache by periodically fetching an I/O rate from the LRU cache or periodically fetching a hit rate from the LRU cache.

The present invention fetches an I/O rate and a hit rate from the LRU cache in order to improve a model of the cache. According to the present invention, the hit rate refers to the number of I/O requests in a given period of time where the data element requested was residing in the cache at the time of the I/O request. The I/O rate is the number of I/O requests sent to the cache by one or more hosts. The I/O rate includes requests that are hits and requests that are misses. The LRU cache calculates a hit rate and I/O rate and provides these rates on request. Specification page 13.

Applying these definitions to *Pirolli*, a hit rate for the invention described in *Pirolli* is the number of document requests in a given period of time where the document requested was residing in cache at the time of the document request. Similarly, an I/O rate for the invention described in *Pirolli* is the number of document requests intercepted by the cache in a given period of time. *Pirolli* contains no teaching regarding the calculation or retrieval of either a hit rate or an I/O rate.

In rejecting claim 7, the Examiner points out that *Pirolli* teaches "determining the documents stored in the cache should be deleted based on the needs list record." However, determining which documents should be deleted is not related to either an I/O rate or a hit rate. The Examiner further notes that *Pirolli* teaches retrieving a context from a document stored in a cache. The context is a set of keywords used to define a query. The query is run against the documents in the cache to determine which documents are closely related to the query. This process determines which documents in the cache are closely related to the query. *Pirolli*, Col. 9, lines 5-28. However, this determination is not related to either an I/O rate or a hit rate for the cache.

Therefore, because *Pirolli* fails to teach or describe fetching an I/O rate and a hit rate from an LRU cache, as required by amended claim 7, Applicant asserts that claim 7 is in condition for allowance over *Pirolli*.

Claim 8

Pirolli does not anticipate claim 8 because *Pirolli* does not disclose modeling a cache by calculating a single reference residency time for a data element within the cache. The present invention teaches computing a Single Reference Residency Time (SRRT) that is an approximation of the average time it takes a data element that has been fetched into the cache to "fall through" or be removed from the cache. The present invention employs a model, which is a representation of the contents of the cache but is not the cache itself. The model uses the SRRT to approximate when a particular data element will be deleted from the actual cache so that the particular data element can also be deleted from the model. Specification page 14.

The Examiner points out that *Pirolli* teaches a history data structure. The history data structure records a vector of the times of day when a client computer requested a particular document. Each time a client computer requests a particular document, the time of the request is recorded in the vector. In addition to the time of the request, a variable storing the number of times the document has been referenced is incremented. The vector and the variable may be used to determine how recently and how frequently the document is referenced. *Pirolli*, Col. 6, line 62 – Col. 7, line 7.

However, the history data structure does not contain a variable equivalent to the SRRT that approximates the time that the document will remain in the cache before being deleted. The history data structure contains information about a particular document during a particular time span. During the time span, the document may have been placed into the cache and deleted from the cache multiple times since there is no guarantee that the document will remain in the cache during the entire time span.

Pirolli further teaches computing a probability that a document will be needed during a day. *Pirolli*, Col. 8, lines 25-32. However, the probability that a document will be requested by a client computer is not the same as an SRRT, which is an approximation of the time that a document, once requested by a client computer, will remain in a cache before being deleted. *Priolli* does not teach computing an SRRT value.

Therefore, because *Pirolli* fails to teach or describe modeling a cache by calculating a single reference residency time for a data element within the cache as recited in claim 8, Applicant asserts that claim 8 is in condition for allowance over the *Pirolli*.

Claim 12

Pirolli does not anticipate claim 12 because *Pirolli* does not disclose a priority value assigned to the preceding data element priority value plus one when the preceding data element is found in the cache.

The present invention teaches assigning a priority value to each data element contained in the model of the cache. As new data elements are added to the model, they are assigned a priority value equal to the preceding data element priority value plus one. Specification page 17. For example, if the preceding data element priority value was 10, the priority value for the new data element would be 11.

Pirolli teaches calculating a need probability for each document. *Pirolli*, Col. 10, lines 12-23. The need probability is a combination of a history factor and a context factor. *Pirolli*, Col. 9, lines 50-55. The context factor is calculated by applying a vector of key words to the document. *Pirolli*, Col. 9, lines 16-28. *Pirolli* contains no teaching that the need probability for a new document will be the sum of the need probability for the preceeding requested document plus one. To the contrary, *Pirolli* teaches that the need probability for each document is calculated without reference to any other document. Instead, the need probability is based on the history factor, which indicates how recently and how frequently the document is accessed, and the context factory, which indicates how closely correlated the document is to a vector of key words. *Pirolli*, Col. 9, lines 50-55.

Therefore, because *Pirolli* fails to teach or describe a priority value assigned to the preceding data element priority value plus one when the preceding data element is found in the cache as recited in claim 12, Applicant asserts that claim 12 is in condition for allowance over *Pirolli.*

Claim 19

Pirolli does not anticipate claim 19 because *Pirolli* does not send an I/O request to the *cache*. As discussed above in relation to claim 1, the present invention sends I/O requests to an LRU cache, not to a memory device. In contrast, *Pirolli* teaches prefetching by sending an I/O request to a *server*, not to an LRU cache. *Pirolli*, Abstract, Col. 2, lines 55-57, Col. 5, line 62 – Col. 6, line 4. Therefore, because *Pirolli* fails to teach or describe sending an I/O request to the cache as recited in claim 19, Applicant asserts that claim 19 is in condition for allowance over *Pirolli.*

Claim 21

Pirolli does not anticipate claim 21 because *Pirolli* does not teach prefetching by requesting data from an LRU cache. As discussed above in relation to claim 19, the present invention sends I/O requests to a LRU *cache*, not to a memory device while *Pirolli* teaches prefetching by sending an I/O request to a *server*, not to an LRU cache. Therefore, Applicant asserts that claim 21 is in condition for allowance over *Pirolli* for at lest the same reasons as claims 1 and 19.

Dependent claims 22-24 depend from independent claim 21. Therefore, Applicant respectfully asserts that dependent claims 22-24 are allowable for at least the same reasons as claim 21.

Applicant has amended independent claims 21 and 29-32 to include substantially the same elements as those added in amended claim 1. Namely, these independent claims recite prefetching of data into an LRU cache and intercepting a request for a data element from a stream of I/O data requests sent from a host and addressed to an LRU cache. Therefore, Applicant respectfully asserts that dependent claims 22-24 are allowable for at least the same reasons as claim 1.

REJECTION OF CLAIMS 13-14, and 25-27 UNDER 35 U.S.C. § 103(a) IN VIEW OF PIROLLI AND DIXON

The Examiner rejected claims 13-14, and 25-27 under 35 U.S.C. § 103(a) as unpatentable over *Pirolli* in view of *Dixon*. Applicant respectfully traverses this rejection.

To establish a *prima facie* case of obviousness, the combination of the prior art references must teach or suggest all the claim limitations. *See* MPEP § 2142. Applicant asserts that a *prima facie* case of obviousness has not been made because, as discussed above, *Pirolli* fails to teach or disclose prefetching of data elements into an LRU cache or intercepting a request for a data element from a stream of I/O data requests passed between a host and the LRU cache as recited in amended claim1. *Dixon* also fails to teach at least these two aspects of the present invention.

Claims 13-14 depend directly or indirectly from the independent claim 1. For the reasons explained above, Applicant respectfully asserts that amended independent claim 1 is allowable. Similarly, claims 25-27 depend directly from the independent claim 21. For the reasons explained above, Applicant respectfully asserts that amended independent claim 21 is allowable. Therefore, Applicant respectfully asserts that claims 13-14, and 25-27 are allowable for at least the same reasons as independent claims 1 and 21. In addition, each of these claims is allowable for the following additional reasons.

Claim 13 recites comparing a priority value of a requested element with a dynamic threshold. The Examiner asserts that the combination of *Pirolli*'s priority value and *Dixon*'s dynamic threshold renders claims 13 and 14 obvious.

Dixon does not teach a priority value. *Dixon* teaches blocks comprising a set of records and prefetching a block based on a threshold comparison. The location of a record within a block is determined and then compared to a threshold. If the location exceeds the threshold then the next adjacent block is prefetched into a cache. The principle of operation of *Dixon*'s threshold is that if the last record requested in a particular operation has a position towards the end of a block, then the next adjacent block is very likely to be requested. Since the next adjacent block is likely to be requested, it is prefetched into the cache. *Dixon*, Col. 17, lines 40-61. The prefetching threshold taught by *Dixon* is expressed as a location within a block.

Claim 13 recites comparing a priority value with a dynamic threshold. The Examiner's proposed augmentation of *Pirolli*'s priority value with *Dixon*'s threshold is improper because *Pirolli*'s priority value has a different basis than *Dixon*'s threshold. *Pirolli*'s priority value may use a scale with a large range to capture a priority. In contrast, *Dixon*'s threshold is based on a location within a block.

There is no reliable transformation between priority values and position within a block since *Pirolli*'s priority is unrelated to the position of a document within a block. *Pirolli* assigns a priority value to each document that attempts to predict how likely it is that the document will be requested. However, position of a document within a block is not used to determine the priority value. Comparing *Pirolli*'s priority value to *Dixon*'s threshold will not result in high priority documents being consistently prefetched because there is no reason that high priority documents will necessarily be above *Dixon*'s threshold.

Thus, comparing *Pirolli*'s priority value to *Dixon*'s threshold would significantly change the principle of operation of *Dixon*'s threshold. If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims prima facie obvious. MPEP §2143.01 p. 2100-132. *In re Ratti*, 270 F.2d 810.

Accordingly, Appellants assert that the Examiner's combination of *Pirolli*'s priority value and *Dixon*'s threshold is improper because it would disable the principle of operation of *Dixon*'s threshold. Therefore, Applicant respectfully asserts that claims 13-14, and 25-27 are allowable for at least this reason.

REJECTION OF CLAIMS 15-18 UNDER 35 U.S.C. § 103(a) IN VIEW OF PIROLLI AND WEINBERGER

The Examiner rejected claims 15-18 under 35 U.S.C. § 103(a) as unpatentable over *Pirolli* in view of *Weinberger*. Applicant respectfully traverses this rejection.

To establish a *prima facie* case of obviousness, the combination of the prior art references must teach or suggest all the claim limitations. *See* MPEP § 2142. Applicant asserts that a *prima facie* case of obviousness has not been made because, as discussed above, *Pirolli* fails to teach or disclose prefetching of data elements into an LRU cache or intercepting a request for a data element from a stream of I/O data requests passed between a host and the LRU cache as recited in amended claim1. *Weinberger* also fails to teach at least these two aspects of the present invention.

Claims 15-18 depend directly or indirectly from the independent claim 1. For the reasons explained above, Applicant respectfully asserts that amended independent claim 1 is allowable. Therefore, Applicant respectfully asserts that claims 15-18 are allowable for at least the same reasons as independent claim 1.

In view of the foregoing, Applicant submits that the application is in condition for immediate allowance. In the event any questions remain, the Examiner is respectfully requested to initiate a telephone conference with the undersigned.

Respectfully submitted,

Brian C. Kunzler
Reg. No. 38,527
Attorney for Applicant

Date: October 28, 2005

KUNZLER AND ASSOCIATES
8 E. Broadway Suite 600
Salt Lake City, UT 84111
Telephone: 801/994-4646

G:\!Client Files\!1200 SanJose\1211\OAR3\1211.OAR3.doc